Bubble Sort

Introduction

Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller or larger elements "bubble" to the top of the list [4].

Performance

In general, the performance of the sorting algorithm is analyzed from two aspects, time complexity and space complexity. Time complexity represents the time taken to execute the algorithm, which is generally considered in three cases: best case, worst case, and average case [6]. Space complexity represents the amount of memory required to complete a program [5].

Use **array** as data structure and **n** denotes the input array size, bubble sort performance is as follows [4]:

Worst-case time complexity	$\boldsymbol{O(n^2)}$
Average time complexity	$\boldsymbol{O(n^2)}$
Best-case time complexity	$\boldsymbol{O(n)}$
Worst-case space complexity	$\boldsymbol{O(1)}$

Pseudocode [3]

Algorithm: BubbleSort(Arr)

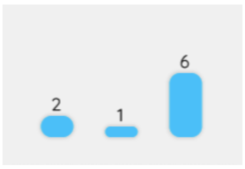
Input: an array of integers Arr

Output: The result of sorting Arr

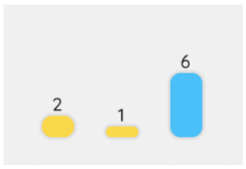
```
length = Arr.length
  for i from 0 to length-1 do
       for j from 0 to length-1-i do
           if Arr[j] > Arr[j+1] then
               swap(Arr[j], Arr[j+1])
         end if
       end for
  end for

  return Arr
```

Example

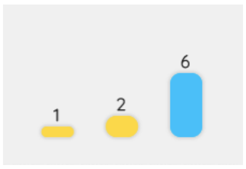
Sorting the list: 2,1,6



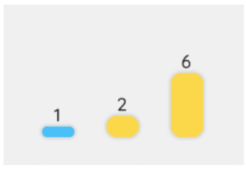
(1)



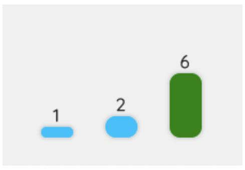
(2)



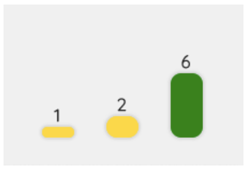
(3)



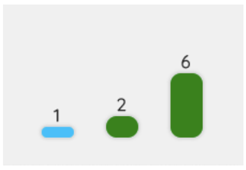
(4)



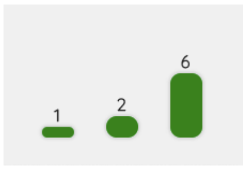
(5)



(6)



(7)



(8)

Implement in programming language

Java[1]

```java
public void bubbleSort(int arr[])
  {
    int n = arr.length;
    for (int i = 0; i < n-1; i++)
      for (int j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
        {
          int temp = arr[j];
          arr[j] = arr[j+1];
          arr[j+1] = temp;
        }
  }
```

JavaScript[2]

```javascript
function bubbleSort(arr) {
    var len = arr.length;
    for (var i = 0; i < len - 1; i++) {
        for (var j = 0; j < len - 1 - i; j++) {
            if (arr[j] > arr[j+1]) {
                var temp = arr[j+1];
                arr[j+1] = arr[j];
                arr[j] = temp;
            }
        }
    }
    return arr;
}
```

C[2]

```c
#include <stdio.h>
void bubble_sort(int arr[], int len) {
  int i, j, temp;
  for (i = 0; i < len - 1; i++)
    for (j = 0; j < len - 1 - i; j++)
      if (arr[j] > arr[j + 1])
      {
        temp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
      }
}
```

References

1. GeeksforGeeks Contributors (2018). *Java Program for Bubble Sort.* [Online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/java-program-for-bubble-sort/ [Accessed 6 Mar. 2021].
2. Runoob Contributors (2018). *Bubble sort*. [Online] Runoob. Available at: https://www.runoob.com/w3cnote/bubble-sort.html [Accessed 6 Mar. 2021].
3. Visualgo Contributors (2014). [Software] Visualgo. Available at: https://visualgo.net/en/sorting [Accessed 6 Mar. 2021].
4. Wikipedia Contributors (2021). *Bubble sort*. [Online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Bubble_sort [Accessed 6 Mar. 2021].
5. Wikipedia Contributors (2021). *Space complexity*. [Online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Space_complexity [Accessed 20 Mar. 2021].
6. Wikipedia Contributors (2021). *Time complexity*. [Online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Time_complexity [Accessed 20 Mar. 2021].